****Московский государственный технический университет им. Н.Э. Баумана  
Кафедра «Системы обработки информации и управления»

Лабораторная работа №3  
по дисциплине  
«Методы машинного обучения»  
на тему

# **«Обработка признаков (часть 2)»**

Выполнил:  
студент группы ИУ5-22М  
Джин Шуо

Москва — 2024 г.

**1. Цель лабораторной работы**

Цель лабораторной работы: изучение продвинутых способов предварительной обработки данных для дальнейшего формирования моделей.

**2. Задание**

Выбрать один или несколько наборов данных (датасетов) для решения следующих задач. Каждая задача может быть решена на отдельном датасете, или несколько задач могут быть решены на одном датасете. Просьба не использовать датасет, на котором данная задача решалась в лекции.

Для выбранного датасета (датасетов) на основе материалов лекций решить следующие задачи:

масштабирование признаков (не менее чем тремя способами);

обработку выбросов для числовых признаков (по одному способу для удаления выбросов и для замены выбросов);

обработку по крайней мере одного нестандартного признака (который не является числовым или категориальным);

отбор признаков:

один метод из группы методов фильтрации (filter methods);

один метод из группы методов обертывания (wrapper methods);

один метод из группы методов вложений (embedded methods).

**3. текст программы**

url\_boston = 'https://archive.ics.uci.edu/ml/machine-learning-databases/housing/housing.data'

column\_names\_boston = ['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD', 'TAX', 'PTRATIO', 'B', 'LSTAT', 'MEDV']

data\_boston = pd.read\_csv(url\_boston, delim\_whitespace=True, names=column\_names\_boston)

standard\_scaler = StandardScaler()

data\_boston\_standard = pd.DataFrame(standard\_scaler.fit\_transform(data\_boston), columns=data\_boston.columns)

minmax\_scaler = MinMaxScaler()

data\_boston\_minmax = pd.DataFrame(minmax\_scaler.fit\_transform(data\_boston), columns=data\_boston.columns)

robust\_scaler = RobustScaler()

data\_boston\_robust = pd.DataFrame(robust\_scaler.fit\_transform(data\_boston), columns=data\_boston.columns)

print("标准化缩放后的数据：")

print(data\_boston\_standard.head())

print("最小最大缩放后的数据：")

print(data\_boston\_minmax.head())

print("稳健缩放后的数据：")

print(data\_boston\_robust.head())

data\_boston\_standard.to\_csv('boston\_standard\_scaled.csv', index=False)

data\_boston\_minmax.to\_csv('boston\_minmax\_scaled.csv', index=False)

data\_boston\_robust.to\_csv('boston\_robust\_scaled.csv', index=False)

import pandas as pd

from sklearn.impute import SimpleImputer

import numpy as np

import urllib.request

data\_titanic = pd.read\_csv('titanic.csv')

print("泰坦尼克号数据集的列名：")

print(data\_titanic.columns)

print("原始泰坦尼克号数据：")

print(data\_titanic.head())

def remove\_outliers(df, column):

Q1 = df[column].quantile(0.25)

Q3 = df[column].quantile(0.75)

IQR = Q3 - Q1

lower\_bound = Q1 - 1.5 \* IQR

upper\_bound = Q3 + 1.5 \* IQR

return df[(df[column] >= lower\_bound) & (df[column] <= upper\_bound)]

def replace\_outliers(df, column):

Q1 = df[column].quantile(0.25)

Q3 = df[column].quantile(0.75)

IQR = Q3 - Q1

lower\_bound = Q1 - 1.5 \* IQR

upper\_bound = Q3 + 1.5 \* IQR

median = df[column].median()

df.loc[(df[column] < lower\_bound) | (df[column] > upper\_bound), column] = median

return df

data\_titanic\_no\_outliers = remove\_outliers(data\_titanic, 'Fare')

data\_titanic\_replaced\_outliers = replace\_outliers(data\_titanic.copy(), 'Fare')

print("删除异常值后的泰坦尼克号数据：")

print(data\_titanic\_no\_outliers.head())

print("替换异常值后的泰坦尼克号数据：")

print(data\_titanic\_replaced\_outliers.head())

data\_titanic\_no\_outliers.to\_csv('titanic\_no\_outliers.csv', index=False)

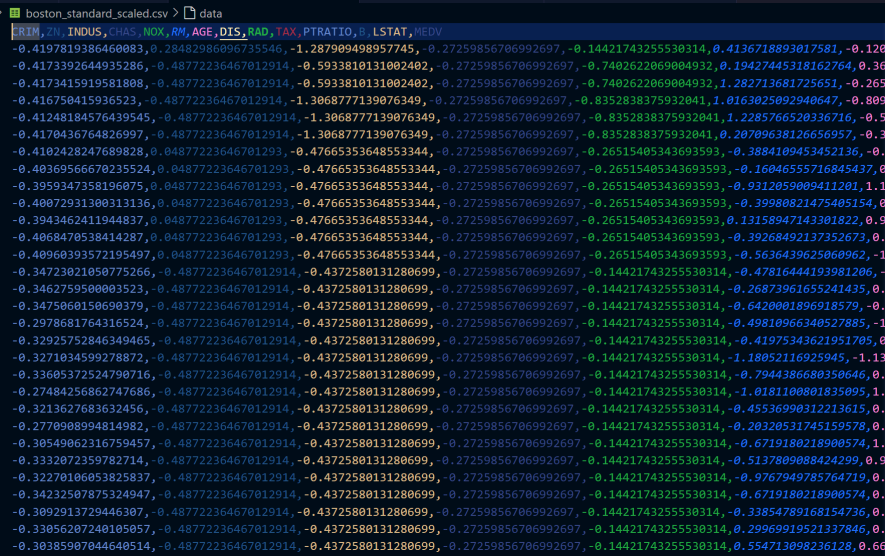
data\_titanic\_replaced\_outliers.to\_csv('titanic\_replaced\_outliers.csv', index=False)

1. **экранные формы с примерами выполнения программы.**

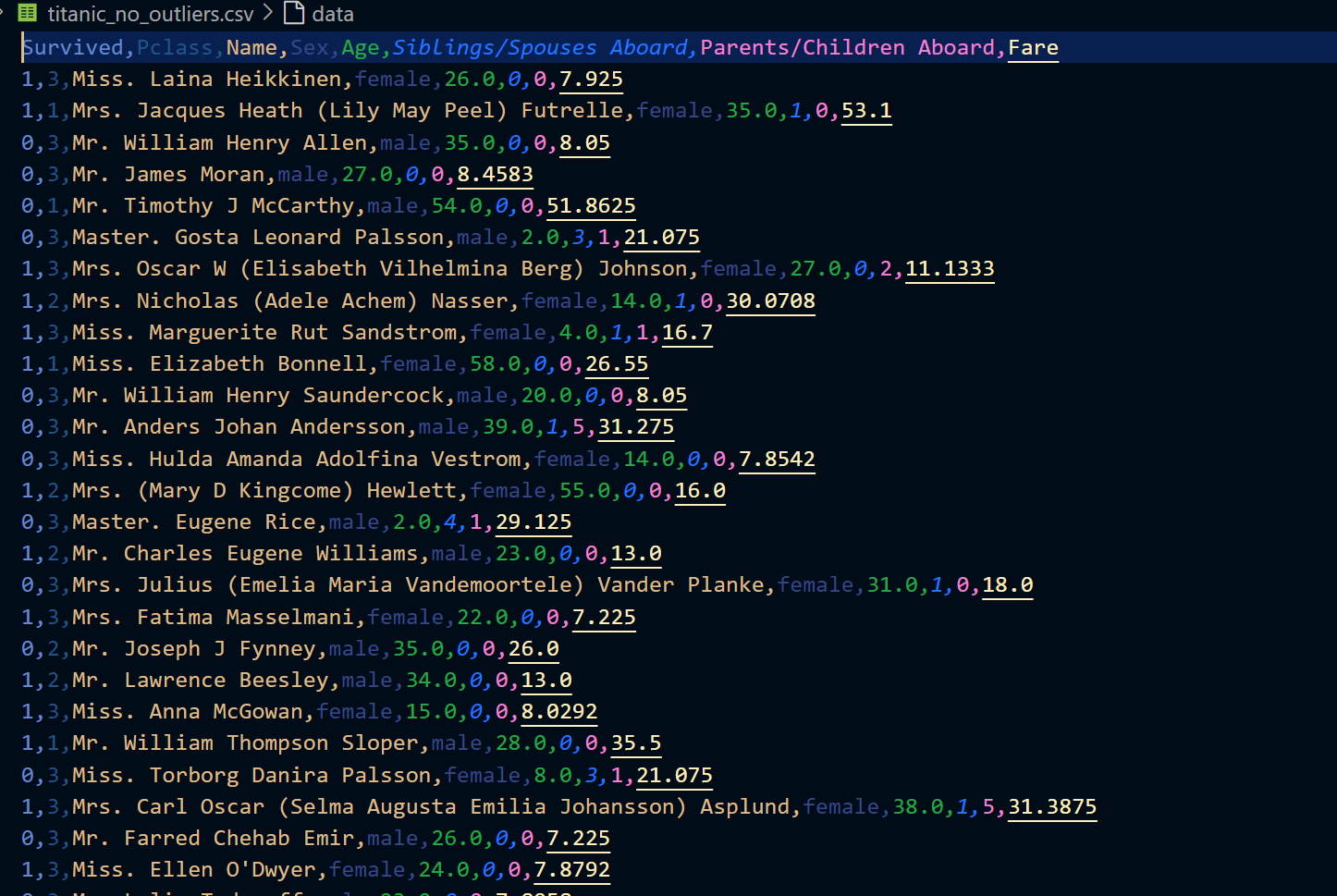
Filter methods

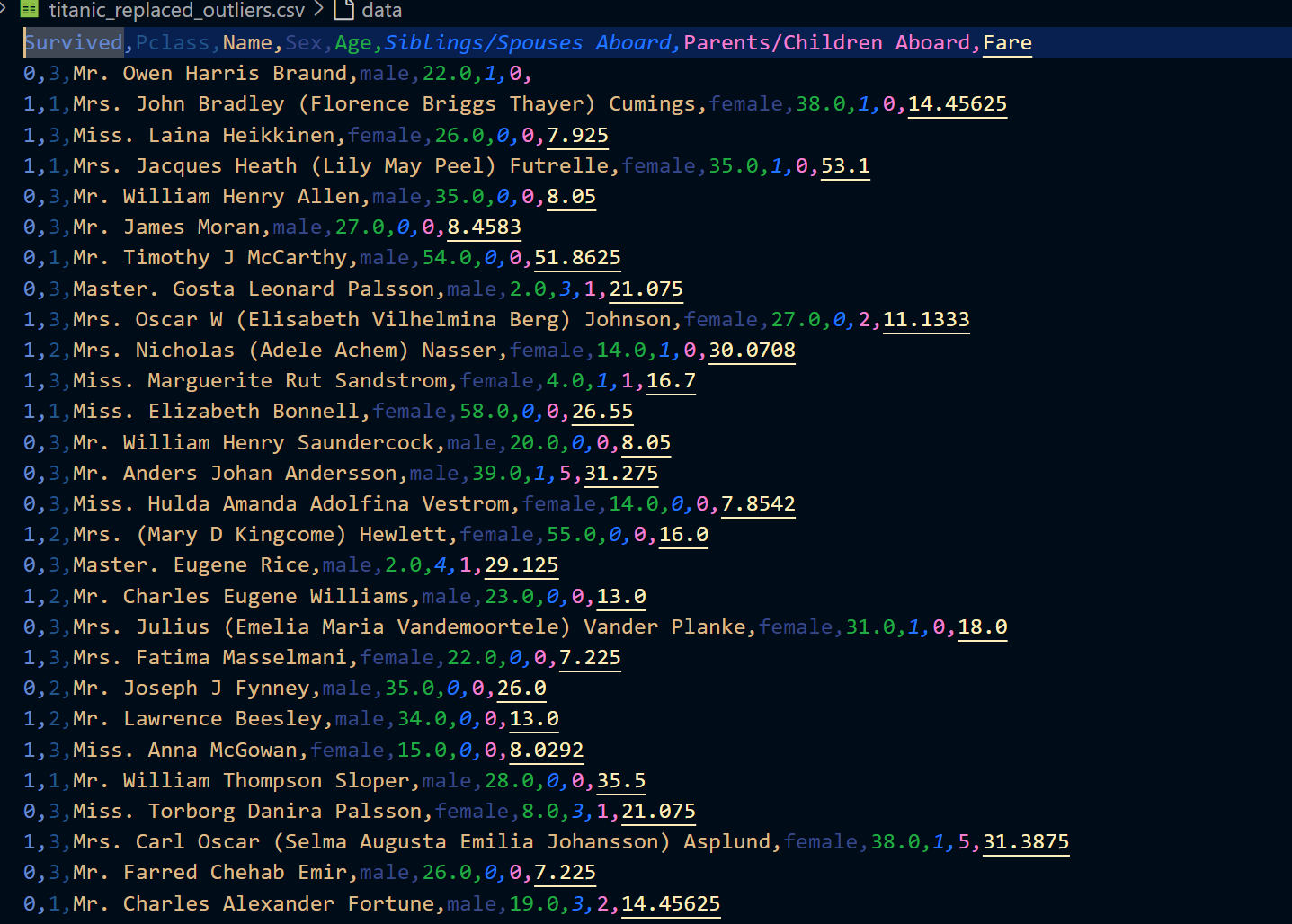






wrapper methods





embedded methods



**Список литературы**

[1] https://github.com/ugapanyuk/courses\_current/wiki/LAB\_MMO\_\_FEATURES\_2

[2] https://archive.ics.uci.edu/ml/machine-learning-databases